



Software Engineering Institute

Process Improvement Should Link to Security: SEPG 2007 Security Track Recap

Carol Woody, PhD

September 2007

TECHNICAL NOTE
CMU/SEI-2007-TN-025

CERT Program

Unlimited distribution subject to the copyright.



CarnegieMellon

This report was prepared for the

SEI Administrative Agent
ESC/XPB
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2007 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Table of Contents

| | |
|--|------------|
| Acknowledgments | v |
| Executive Summary | vii |
| Abstract | ix |
| 1 Process Improvement Should Link to Security | 1 |
| 1.1 Selected Panel Questions | 1 |
| 1.2 Panel Resources | 1 |
| 1.3 Panel Expert Presentations | 3 |
| 1.3.1 Question 1: Getting Credit for Effective Security Processes | 3 |
| 1.3.2 Question 2: Processes for Determining Security Requirements | 4 |
| 1.3.3 Question 3: Measuring Security Processes and Improvement Efforts | 5 |
| 1.3.4 Question 4: Development Processes Contributing to Operational Resiliency | 5 |
| 1.3.5 Question 5: Leveraging Process Improvement for Security in the SDLC | 5 |
| 1.4 Audience Feedback To Panelists | 7 |
| 2 Security Track Presenters Connect Security to Process | 10 |
| 2.1 Security Track Speakers Covered A Range of Security Issues | 10 |
| 2.2 Software Security— Setting the Stage | 10 |
| 2.3 Insider Threats in the SDLC | 12 |
| 2.4 Engineering Safety- and Security-Related Requirements for Software-Intensive Systems | 14 |
| 2.5 Focus on Resiliency: A Process Improvement Approach to Security | 15 |
| 2.6 Getting Started with Measuring Your Security | 16 |
| 3 Strengthening Ties between Process and Security | 18 |
| 3.1 Security Birds of a Feather (BOF) at SEPG07 | 18 |
| 3.2 NDIA Systems Assurance Guidebook | 18 |
| 3.3 DHS Software Assurance Program | 18 |
| 3.4 ISSEA Systems Security Engineering CMM | 19 |
| 3.5 ISO/IEC 15026 “Systems and Software Assurance” | 19 |
| Bibliography | 21 |

List of Figures

Figure 1: Vulnerabilities Reported to CERT Continue to Increase

11

Acknowledgments

While many claim to want security, few are willing to put the time and energy toward its realization. Sincere thanks to the speakers and panelists who supported the Software Engineering Process Group (SEPG) Conference 2007 security track, through sharing of their knowledge and experience. Their expertise and strong interest in making an impact in this area made my role as the coordinator for the security track at the conference both interesting and edifying. The challenges for security are many and each participant provided valuable input toward building an understanding of the needs and available solutions.

The following individuals spoke at SEPG 2007 in the Security Track:

Dawn Cappelli, Software Engineering Institute (SEI)

Donald Firesmith, SEI

Eileen Forrester, SEI

Joe Jarzombek, Department of Homeland Security (DHS)

Nancy Mead, SEI

Michele Moss, Booz Allen Hamilton

Margaret Nadworny, Motorola

Riley Rice, Booz Allen Hamilton

Kenneth R. van Wyk, KRvW Associates, LLC

Lisa Young, SEI

Executive Summary

The security track is a recent addition to the Software Engineering Process Group (SEPG) Conference and attendance has been limited despite the fact that security is on the high-priority list for most organizations these days. For SEPG 2007, speakers were selected to emphasize the connections of security to process and to present as wide a range of viewpoints as time would allow. In addition, a panel discussion was added to the agenda as a way to initiate conversation with attendees, so as to glean their perspectives on the critical issues and problems of linking process and security.

Attendance at the security track sessions for 2007 was much greater than in previous years. Extensive informal feedback from speakers and audience participants indicated a need to share the ideas and discussions with a broader audience than those able to attend. This technical note is one step in the efforts underway to draw a broader audience to those conversations.

The panel provided a forum for the audience to air its concerns and interests. Panel leadership came from the Department of Homeland Security, Motorola, Booz Allen Hamilton, and CERT[®] Coordination Center of the Carnegie Mellon[®] Software Engineering Institute (SEI). Greater detail is provided in Section 1 of this document. The following are selected audience responses this author viewed as key:

- Process is the security enabler—getting the right people at the right place at the right time.
- Security is a separate discipline that must collaborate with existing process areas, but should not be assumed to fully blend with existing processes.
- Security must be a part of the normal organizational information flow and not an add-on when it suits.
- In general there is a lot of awareness of the need to do something but little understanding of how to go about it effectively.
- There is no need for more standards and regulations—there is a great need to implement what we already have throughout the life cycle.
- Evidence that an organization is meeting a security standard should be part of the process measurement.
- Existing practices support security but do not promote it.
- Unless security capabilities are assessed, the associated security processes and practices will not be improved and will not meet the needs of today's environment.

[®] CERT and the CERT Coordination Center are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

[®] Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

In addition to the panel discussion, presentations on the following topics were provided for conference attendees to expose them to the breadth of security issues their organizations should be considering:

- the challenges of security issues
- research linking insider illegal actions to poorly managed system development life-cycle processes
- insight into ways that safety and security engineering provide related requirements in software-intensive systems
- an improvement approach to benchmarking an organization's operational resiliency
- initial steps to measuring security

A summary of the linkages of each presentation to process issues is provided in Section 2.

Section 3 describes the steps underway to strengthen the ties between security and process.

Abstract

Security is a very visible issue these days for software. New software products are continuously reported to be vulnerable to attack and compromise; organizations must support an expensive unending update-and-upgrade cycle. Process improvement has been proposed as a mechanism for addressing security challenges, but the Capability Maturity Model Integration (CMMI[®]) approach does not specifically address security, so the linkages for the Software Engineering Process Group (SEPG) community are unclear. The security track at the SEPG 2007 conference was developed to provide a forum for identifying the appropriate ties between process improvement and security. This document summarizes the content shared at the conference and identifies several subsequent steps underway toward strengthening those ties.

1 Process Improvement Should Link to Security

For several years, the security track at the Software Engineering Process Group (SEPG) Conference has had limited attendance compared to other conference tracks. Since security is a critical issue for so many organizations internationally, the reason for lack of interest has been difficult to determine. To identify reasons for such limited interest and to establish a plan to appropriately position this critical topic for future conferences, a panel of experts from a range of disciplines was convened to speak to aspects of this potentially large problem space and to solicit audience response.

1.1 SELECTED PANEL QUESTIONS

The following questions were assembled by senior technical staff from the Carnegie Mellon® Software Engineering Institute (SEI) security and process programs to guide the conversations and encourage audience response:

1. What must be done for organizations to receive credit for responsible development processes when it comes to evaluating security? How can the security-process improvement connections be made clear enough to warrant credit and in what ways would credit be meaningful?
2. How do you currently determine security requirements and what process improvement approaches would help you to do a better job?
3. What do you need in order to start measuring processes and improvement efforts for their effect on security?
4. How can responsible development processes contribute to operational resiliency?
5. What would promote an organization's ability to leverage process improvement capabilities to address security throughout the software development life cycle (SDLC)?
6. In what ways should process improvement support security in the SDLC?

1.2 PANEL RESOURCES

For each question a panel expert proposed key issues and solicited audience input to expand or dispute his or her assertions.

Question # 1 Panel Expert: Margaret Nadworny, Motorola

Question # 2 Panel Expert: Nancy Mead, SEI

Question # 3 Panel Expert: Michele Moss, Booz Allen Hamilton

Question # 4 Panel Expert: Lisa Young, SEI

Question # 5 Panel Expert: Joe Jarzombek, DHS

Question # 6 Facilitator (Eileen Forrester) led audience discussion

® Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

About Margaret Nadworny

Margaret Nadworny was Managing Director for the first organization to be assessed at the SEI Capability Model[®] (CMM[®]) level 5 (L5) in Russia, and Managing Director for the first organization to be assessed at SEI CMM L5 in Poland. She currently holds the position of Director of Security Awareness for worldwide software development organization with a high percentage of CMM and Capability Maturity Model Integration (CMMI[®]) L5 staffing. Her organization, Motorola Software Group and Motorola, are culturally aligned with CMM/CMMI. Nadworny is responsible for incorporating security into CMM/CMMI strategic efforts for improving security and quality in Motorola Products.

About Nancy Mead

Nancy R. Mead is a senior member of the technical staff in the Networked Systems Survivability Program at the SEI. The CERT[®] Coordination Center is a part of this program. Mead is also a faculty member in the Master of Software Engineering and Master of Information Systems Management programs at Carnegie Mellon University. Her research interests are in the areas of information security, software requirements engineering, and software architectures.

Mead has more than 100 publications and invited presentations. She is a Fellow of the Institute of Electrical and Electronic Engineers, Inc. (IEEE) and the IEEE Computer Society and is also a member of the Association for Computing Machinery. Mead received her PhD in mathematics from the Polytechnic Institute of New York, and received a BA and an MS in mathematics from New York University.

About Michele Moss

Michele Moss is a security engineer with more than 12 years of experience in process improvement. She has assisted numerous organizations in maturing their information technology, information assurance, project management, and support practices through the use of the capability maturity models including the CMMI and the Systems Security Engineering Capability Maturity Model (SSE-CMM). She specializes in integrating security processes and practices into project life cycles. Moss is an active member of the Systems Security Engineering Community. She is a Certified Information System Security Professional (CISSP), an active member of the International Systems Security Engineering Association (ISSEA), and the Co-Chair of the DHS Software Assurance Working Group on Processes and Practices. Moss has spoken on the subject of integration of information security into SDLC and information security measurement at numerous conferences, including the SEPG, National Defense Industrial Association (NDIA), Systems and Software Technology (SSTC), and ISSEA confer-

[®] The Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

[®] CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

[®] CERT and the CERT Coordination Center are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

ences. Moss has also taught classes on the subject of information security process improvement.

About Lisa Young

Lisa Young is a senior member of the technical staff on the Survivable Enterprise Management team of the Networked Systems Survivability (NSS) Program at the SEI. At present, Young is part of a team that is developing a framework for security process improvement called the CERT Resiliency Engineering Framework.

Young teaches the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE[®]) risk-based security assessment methodology at the SEI. She holds the designations of Certified Information Systems Auditor (CISA) and CISSP. Young has over 25 years experience in information technology in the areas of IT governance, information audit and security, and risk management.

About Joe Jarzombek

Joe Jarzombek is the Director for Software Assurance in the Department of Homeland Security (DHS) National Cyber Security Division. He leads government interagency efforts with industry, academia, and standards organizations to shift the security paradigm away from patch management. He does this by addressing the need for security education and training in the work force, more comprehensive diagnostic capabilities, and security-enhanced development and acquisition practices.

After retiring from the U.S. Air Force as a Lt. Colonel in program management, Jarzombek worked in the cyber security industry, directing product and process engineering. He later served in two software-related positions within the Office of the Secretary of Defense prior to accepting his current DHS position. Throughout his career he has actively led process improvement initiatives; this includes serving on the CMMI Product Development Team and later on the CMMI Steering Group. He also co-led an effort to integrate safety and security into integrated CMMs. As a Project Management Professional, Jarzombek has spoken extensively on measurement, software assurance, and acquisition topics. He encourages further stakeholder involvement in software assurance efforts via interagency software assurance forums, working groups, and the Build-Security-In Web site.

1.3 PANEL EXPERT PRESENTATIONS

1.3.1 Question 1: Getting Credit for Effective Security Processes

To obtain credit for responsible development processes supporting security, Nadworny recommends that organizations build on the development processes for quality that are already in the capability maturity model: organizational policies, training plans, senior management support, supplier management, and project management. In addition, Nadworthy's organization, Motorola, has established five additional process areas:

[®] OCTAVE is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

- secure development processes (e.g., threat modeling)
- secure management processes (e.g., security in commercial off-the-shelf and open source products)
- organization security processes (e.g., security metrics reviewed and actions taken)
- discovery of security vulnerabilities and risks (e.g., use of static analysis tools)
- corrective actions (e.g., root-cause analyses of vulnerabilities)

Today's environment demands secure development, but without good baseline information it is difficult to establish organizational return on investment for security. By giving credit to progress made in securing the life cycle, organizations will have a vehicle for enhancing their justification efforts. Since CMMI already has a mechanism to evaluate organizations and many of the existing processes are critical to security, it makes more sense to extend the current model rather than apply a new one. Potentially separate ratings for non-security process areas and secure process areas are needed so security does not "get lost" in the overall summary.

1.3.2 Question 2: Processes for Determining Security Requirements

Security requirements are provided by clients and developed by requirements or software engineers using a documented process. Requirements management may be implemented with or without underlying requirements engineering processes.

Security requirements are frequently an afterthought, selected from standard mechanisms, such as firewalls or virus detection software, or copied from standard lists, and often given low priority relative to functional requirements.

Process improvement approaches would help organizations do a better job of identifying security requirements. Use of a specific security requirements engineering process, such as SEI Security Quality Requirements Engineering (SQUARE), would provide needed structure. Organizations need to do a better job of risk analysis, which process improvement will support. Specifically, scheduled security requirements elicitation sessions would provide a needed spotlight on this frequently neglected area. Also, organizations should consider incorporating security requirements into requests for proposals, and proposal responses.

SQUARE is structured in the following steps:

1. Identify business goal.
2. Identify security goals.
3. Develop artifacts to support security requirements definition.
4. Assess risks.
5. Select elicitation technique(s).
6. Elicit security requirements.
7. Categorize requirements.
8. Prioritize requirements.
9. Inspect requirements.

Additional information about SQUARE is available at <http://www.cert.org/sse/square.html>.

1.3.3 Question 3: Measuring Security Processes and Improvement Efforts

Measurement can provide insights into many aspects of software and system development: project management, compliance, assurance, return on investment (ROI), and process management are a few. A measurement infrastructure must be in place to enable and facilitate measurement. A solid measurement infrastructure requires the following: processes and procedures to be defined, documented, and eventually institutionalized; data that is available and easily collected; and a measurement process that is an integral part of every business process.

Process implementation evidence that results from integration of the security life cycle into the SDLC creates tangible data that can be leveraged to support implementation, efficiency and effectiveness, assurance, and ROI measurements. Applying capability maturity models enhances the ability of an organization to reliably collect and use security measurement in a consistent and repeatable manner. Process Appraisals against capability maturity models such as CMMI and ISO/IEC 21827 (SSE-CMM) provide insight into the process maturity of processes and practices. An integrated appraisal of the CMMI and SSE-CMM with a result of level 2 or higher indicates that processes and practices are in place to support the integration of information assurance (IA) into the life cycle. Additionally, at level 2 basic infrastructure exists to facilitate measurement of security.

1.3.4 Question 4: Development Processes Contributing to Operational Resiliency

Operational resiliency is the organization's ability to sustain its mission in the face of operational risks such as failed internal processes, inadvertent or deliberate actions of people, problems with systems and technology, and external events. It emerges from the coordination and execution of security, business continuity, and IT operations management toward a common goal. In practice operational resiliency requires a focus on the business processes and the linked people, information, facilities, and technology needed to keep the operational capacity of the organization from disruption.

Young noted that security is too frequently viewed as a technical problem and addressed as an afterthought with poorly defined operational practices [Young 2007]. Instead, security must be established as a business issue owned by the organization and evaluated as an investment. Security should be positioned as an enterprise process that is measured and managed for its contribution to attaining and sustaining operational resiliency. The CERT Resiliency Engineering Framework (see Section 2.5 for additional information) is a process improvement model for resiliency engineering [Caralli 2007]. It is built on effective process improvement of enterprise capabilities and the integration of protection and sustainment for critical business services through resiliency engineering.

1.3.5 Question 5: Leveraging Process Improvement for Security in the SDLC

Trustworthy software systems require an effective synthesis of component technology, quality of service for availability and reliability, and performance, correctness, safety, privacy, and security. A holistic approach is needed to factor in all relevant technologies, protection initiatives, and contributing disciplines to achieve this result. Standards provide a basis for certification. Enabling

technologies and life-cycle processes must be used effectively at all layers within the software supply chain to achieve a trustworthy composition.

Predictable execution is a key component of trustworthy software systems. Each process within the life cycle contributes to a dependable, trustworthy outcome:

- Development/acquisition practices and process capabilities establish the criteria for assuring integrity and mitigating risks—the organization must know what it takes to obtain what it wants.
- Requirements engineering, threat modeling and analysis, failsafe design and defect-free code, and supply chain management contribute to building and/or acquiring what we want.
- Production assurance evidence, comprehensive testing and diagnostics, formal methods, and static analysis contribute to our understanding of what has been built and/or acquired.
- Policies and practices for use and acquisition, composition of trust modeling, and hardware support allow us to use what we have obtained based on a comprehensive understanding of its capabilities and limitations.

Development processes based on sound practices, standards, and practical guidelines for development of secure software are a major component of the DHS Software Assurance Program. Processes alone are not sufficient. Both developers and users must be trained. Technology is needed for diagnostic tools and measurement. Acquisition must support trustworthy development through due-diligence questions, specifications, and guidelines for acquisition and outsourcing.

Process improvement should also be leveraged to contribute to system and software assurance by providing evidence that:

- An infrastructure for safety and security is established and maintained.
- Safety and security risks are identified and managed.
- Safety and security requirements are satisfied.
- Activities and products are managed to achieve safety and security requirements and objectives.

In September 2004, the report, *Safety and Security Extensions for Integrated Capability Maturity Models* was released, reflecting two years of interagency and industry work in synthesizing and harmonizing practices from four safety standards and four security standards that could be used with CMMs [Ibrahim 2004]. See

http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf

In an era riddled with asymmetric cyber attacks, claims about system reliability, integrity, and safety must include provisions for built-in security of the enabling software. If organizations claim to have the capabilities to deliver dependable and trustworthy products and services, then they must have their processes and practices assessed relative to security.

1.4 AUDIENCE FEEDBACK TO PANELISTS¹

Process is the security enabler – getting the right people at the right place at the right time.

Security must be a part of the normal organizational information flow as well as separated at key points to support the focused use of individuals such as the DAA (designated approval authority).

Combining safety and security such as UK has done for accreditation and certification can relieve the burden on developers if the right resources can be at the right place at the right time (experts are needed to address security and safety – these cannot be effectively “added” to existing resources)

Security is a separate discipline that must collaborate with existing process areas but should not be assumed to fully blend with existing processes.

Must combine security into the other processes otherwise it is not done. The omission is no penalty at the development level since the problems do not show up until after implementation.

Need for specific guidelines and standards to provide a level of “goodness” which is missing from current CMMI – similar problem to quality since can have excellent processes but this says nothing about the quality. Process focus alone is not sufficient for security – that does not address the problems of vulnerable products.

A requirements methodology is part of the process now, but does not specifically address security which requires specific artifacts.

Security requirements (as part of the non-functional requirements) are given a lower priority than functional features – OEMS ask for adherence to ISO standard but require outside forces (such as regulation) to push for specific security needs.

Federal government including the Department of Defense (DoD) has very specific regulations from the Office of Management & Budget (OMB) which are escalating in importance rapidly and will require compliance long before SEI changes the CMMI.

Security is like trying to hit a moving target – more problems surface daily as attackers improve skills and technology opportunities change. As with any other requirements, the contract is awarded and six months later the regulations change.

The integrator of the range of products assembled to define a delivered implemented product (software, system, etc.) has to take the responsibility for creating the derived security requirements that must be drilled down to each participant in the construction – this series of responsibilities is not defined and does not happen today so security is an after-thought when things don’t work at integration time (a highly expensive point to learn about a problem).

Eliciting security requirements should be done earlier but few users (or product customers) understand their security needs to be able to articulate them.

¹ Except for a few minor edits for intelligibility; audience feedback appears as originally submitted.

There is lots of awareness of the need to do something but little understanding of how to go about doing something effective.

Appraisal teams will need security trained participants to provide required competence in the SCAMPI assessment if security capabilities are included. The focus of SCAMPI on four projects may be insufficient to fully understand an organization's security capabilities; additional sampling may be required.

Security must be considered at all lifecycle phases since it is very hard to separate out security and have any effective outcome.

Development expertise has to be familiar with the operational environment of the system to define appropriate measurement – operational security expertise must be brought into the process earlier since the operational context is where systems break.

Security must be positioned as an investment and not an expense – all attendees noted these costs are expensed at this point. The costs are applied at the project and system level when they should be considered at the enterprise level.

Predictable execution is another way of approaching security within development that may give it more visibility; the ability for people unknown to the enterprise to remotely access and change systems and applications adds a dynamic that requires security covering the full flow of information from start to finish of a transaction.

There is a need for greater diagnostic capability to determine security needs – development scope is too limited to effectively evaluate the range of threats that should be considered based on the range of uses software and systems are applied in today's connected environment. Developers are unaware that the organization must manage and control the supply chain. Software developed by others (either via open source, libraries, or reuse of legacy software) must be evaluated using security criteria.

Consumers must demand better security that can resist and recover from common misuse patterns such as those enumerated in the common attack patterns (CAPEC) and common weakness enumeration (CWE) repositories.

Much of security guidance and standards is process agnostic. Evidence that an organization is meeting a standard should be part of the process of measurement – critical touch points.

Specifying just how secure a system needs to be (what is good enough) should be based on what assets are to be protected and not the known frequency of attacks.

Do not have good definition of what is “good enough” security.

Regulatory references are impossible to distill to each individual project (too costly) and an organization should have a broader process in place to make the delegations to each individual project. Organizational process should be in place to support development in this area of distillation for consistency of application.

Developers are told to “deploy anyway” in the face of known security gaps based on cost and schedule demands. When does security become a strong motivator for decision-making?

There is no need for further standards and regulations. There is a need to pick a set and fully and effectively implement it throughout all of the development life cycle. The current range of options leaves organizations picking and choosing with no evaluation mechanism for effectiveness.

2 Security Track Presenters Connect Security to Process

2.1 SECURITY TRACK SPEAKERS COVERED A RANGE OF SECURITY ISSUES

The following presentations were provided for attendees at the conference. A summary of the key issues addressed by each speaker and the critical links of their focus areas to process improvement is provided in the remaining sections of this document:

- A tutorial introduced SEPG attendees to the challenges of security issues. Kenneth van Wyk, a security consultant and author of *Secure Coding Principles and Practices*, shared his experiences and insights in a presentation titled “Software Security – Setting the Stage” [van Wyk 20007].
- Dawn Cappelli from CERT shared recent research linking insider illegal actions to poorly managed system development life cycle processes in her presentation titled “Insider Threats in the SDLC” [Cappelli 2007].
- Insight into ways that safety and security engineering provide related requirements in software-intensive systems was presented by Don Firesmith, from the Acquisition Support Program at SEI. His presentation was titled “Engineering Safety- and Security-Related Requirements for Software-Intensive Systems” [Firesmith 2007].
- “Focus on Resiliency: a Process Improvement Approach to Security,” which provided an improvement approach to benchmarking an organization’s operational resiliency, was presented by Lisa Young from CERT [Young 2007].
- Michele Moss and Riley Rice from Booz Allen Hamilton presented “Getting Started with Measuring Your Security.” An earlier version of this presentation is available [Moss 2006].

2.2 SOFTWARE SECURITY— SETTING THE STAGE

This section provides a summary of the major points provided by Kenneth van Wyk in his SEPG tutorial [van Wyk 2007]. The security problem is reinforced by the continuous growth of software vulnerabilities that more than double each year, as demonstrated by CERT research. The graph in Figure 1 vividly reinforces the challenge that has no expected stopping point in the near term.

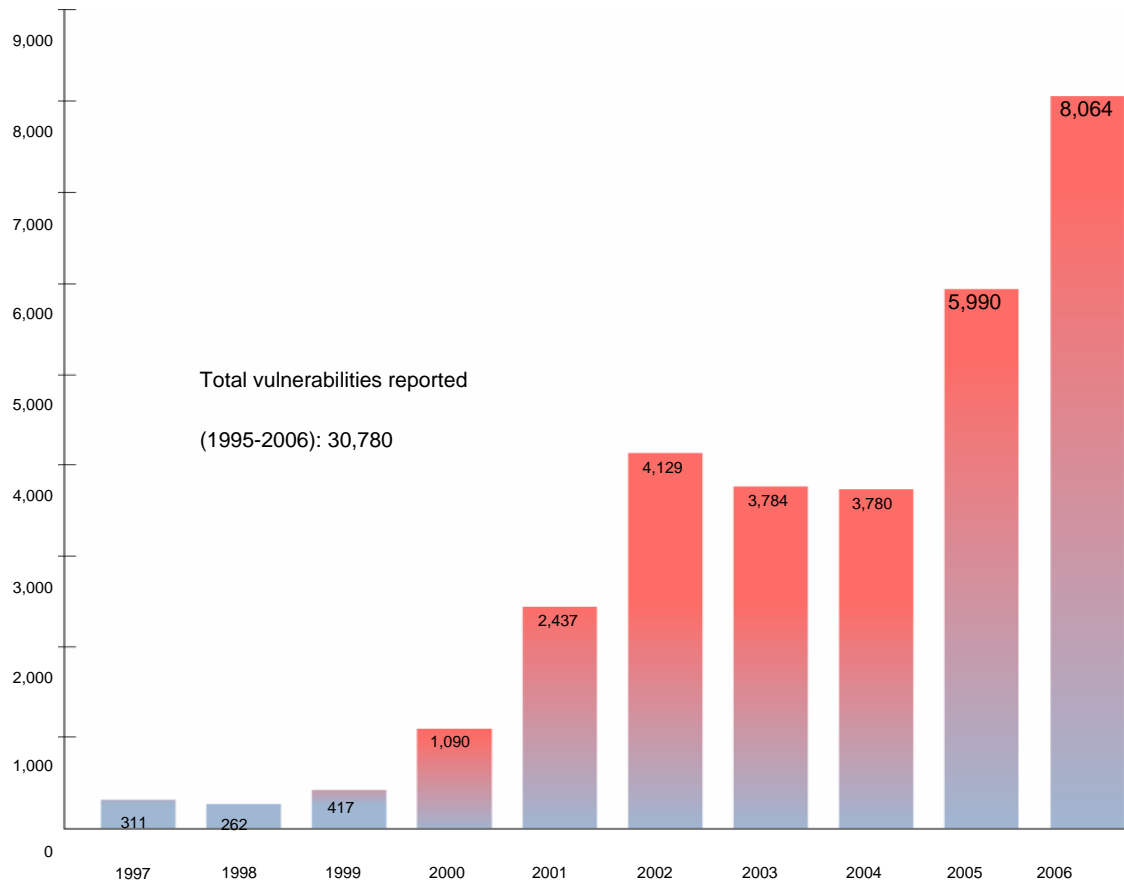


Figure 1: Vulnerabilities Reported to CERT Continue to Increase

According to van Wyk, changes to the vulnerability growth trend to which software contributes as a result of poor quality and inattention to security issues will require attention to the following key areas:

- Consumers must demand more security.
- Software developers must develop a stronger knowledge of vulnerabilities, attacks, and threats.
- IT security must develop a stronger understanding of software development.
- Failures for security must be studied and carefully analyzed as in other engineering disciplines.
- Business risk must drive the decisions of technology including security.
- Perimeter controls and hardware solutions must not be considered adequate protection for software.
- Software testing and penetration testing must not be considered sufficient for security.
- What can go wrong must be considered as thoroughly as are functional needs.
- Security must not be always saying “no” and thus regarded as an impediment to the organization [van Wyk 2007].

Software security problems are complicated. At a high level there are two major categories: (1) implementation bugs such as buffer overflows, race conditions, and untrustworthy input problems and (2) architectural flaws such as protection failures, misuse of cryptography, broken access control, along with design compartmentalization and fragility. Software security is about building things properly. Tools to identify bugs are improving, but popular languages such as C and C++ do not enforce secure coding. Widely used communication protocols are weak; one example is the 802.11b WEP protocol which contains well-documented design flaws that allow easy circumvention of encryption. Developers must learn from past mistakes so as to better write secure code. They must consider available examples of configurations, scripts, design flaws, and buggy code to avoid repeating mistakes.

In order to understand how to build secure software, the developer must understand how an attacker goes about breaking code. Below are the best practices that should be incorporated at critical points within the development life cycle (touch points):

- Clearly describe the security needs of functional requirements such as protection requirements for critical data elements and user authentication.
- Define abuse cases to describe scenarios of important non-normative behavior that should not be allowed to occur. Build an attack model from possible attack patterns, requirements, and use cases to clarify what should not be allowed to happen (anti-requirements).
- Review code with a tool that detects security vulnerabilities to eliminate obvious errors.
- Perform an architectural risk analysis to consider resistance to attack, design ambiguity, and design weakness (fragility). Prioritize identified risks for mitigation.
- Use penetration testing to evaluate the operational severity of risks identified in risk analyses.
- Perform risk-based security testing using the abuse cases and architectural risk review to identify critical test scenarios. Concentrate on ensuring that security breaks won't occur.
- Involve operational security resources to apply their knowledge and provide for external review to make sure obvious vulnerabilities are not missed.

Even applying a few relatively simple precautions can make a tangible difference in moving along the path toward secure software.

2.3 INSIDER THREATS IN THE SDLC

Dawn Cappelli presented lessons learned from actual incidents of fraud, theft of sensitive information, and IT sabotage based on research from an insider threat study [Cappelli 2007]. Actual cases were examined for technical and psychological aspects to develop information for prevention and early detection. An *insider* in the context of the threat study is defined as a

current or former employee or contractor who intentionally exceeds or misuses an authorized level of access to networks, systems, or data in a manner that harms a specific individual or negatively affects the security of the organization's data, systems, and /or daily business operations.

Of the 116 cases reported, 54 were IT sabotage, 44 were fraud, and 40 involved theft of information. As a result of this insider misuse, organizations experienced several types of impact. Some companies went out of business and others sustained fraud losses of up to \$691 million. Driver's licenses were created for individuals who could not legally obtain them, a telecommunications firm experienced disruption of communication services, critical data such as court records and credit records were inappropriately modified, and customer systems were infected with viruses.

The research identified a range of areas across the system development life cycle where neglect of good processes and practices provided an increased opportunity of an insider security attack.

- Requirements definition oversights
 - Neglecting to define authentication and role-based access control requirements allowed easier insider attacks.
 - Neglecting to define security and separation of duties for automated business processes simplified attack methods for the insider.
 - Neglecting to define requirements for automated data integrity checks reduced the possibility of detection of improper actions.
- System design oversights
 - Insufficient attention to security details in automated workflow processes enabled malicious activity.
 - Insufficient separation of duties and lack of oversight for system actions facilitated insider crime.
 - Neglecting to consider security vulnerabilities posed by built-in authorized system overrides provided ways for insiders to bypass what controls were included in the system.
- System implementation exploits
 - Lack of code reviews allowed insertion of “backdoors” in the source code.
 - Inability to attribute actions to a specific user enabled a project manager to sabotage a team of developers when project deadlines could not be met.
- System deployment oversights
 - Lack of enforcement of documentation practices and backup procedures prohibited recovery efforts when production source code was deleted by an insider.
 - Use of the same password file for development and production enabled insiders to steal data from the operational system.
- System maintenance issues
 - Lack of code reviews facilitated insertion of malicious code.
 - Ineffective configuration control practices enabled release of unauthorized code into production.
 - Ineffective or missing backup processes amplified the impact of a mass deletion of data.
 - Inappropriate access to source code by end users allowed them to modify security measures that limited their capabilities.
 - Ignoring known system vulnerabilities provided exploit opportunities for systems only accessible within the organization.

Effective and well-managed processes for access control, source code control, and configuration management are needed to reduce the risk of sabotage and information theft. Formal code inspections and well-structured automated workflows, which include appropriate separation of duties and validation, are needed to reduce the risk of sabotage and fraud.

The *Cylab Common Sense Guide* (available at http://www.cert.org/insider_threat) includes the following best practices that should be incorporated into organizational processes and practices to appropriately address insider threat [SEI CERT 2007]:

- Institute periodic enterprise-wide risk assessments.
- Institute periodic security awareness training for all employees.
- Enforce separation of duties and least-privilege limitations.
- Implement strict password and account management policies and practices.
- Log, monitor, and audit employee online actions.
- Use extra caution in authorizing system administrators and privileged users.
- Actively defend against malicious code.
- Use layered defense against remote attacks.
- Monitor and respond to suspicious or disruptive behavior.
- Deactivate computer access following termination.
- Collect and save data for use in investigations.
- Implement secure backup and recovery processes.
- Clearly document insider threat controls.

2.4 ENGINEERING SAFETY- AND SECURITY-RELATED REQUIREMENTS FOR SOFTWARE-INTENSIVE SYSTEMS

Don Firesmith described the relationship among the disciplines of safety engineering, security engineering, and requirements engineering [Firesmith 2006]. These three critical areas are treated too frequently as unrelated and independent functions within the development of software-intensive systems, but should be closely aligned to deliver effective reduction of risk of unauthorized harm.

Each discipline differs in training, resources, growth paths, job titles, underlying concepts and terminologies, tasks, tools, and techniques. Requirements, safety, and security also engage separate processes, resulting in inefficiency, duplication of effort, and gaps in the resulting products.

It's important to address the limitations of current development processes in their analyses of requirements. More than half of all project failures resulting from cost overruns, schedule overruns, major undelivered functionality, project cancellation and unused delivered systems can be linked directly to poor and insufficient requirements. Poor requirements also have been identified as a major root cause of many accidents involving software-intensive systems.

Greater linkages between safety and security requirement analysis can lead to improved consistency, reuse of techniques, and a reduction in unnecessary overlap and redundant work. Instead of

separating safety and security into separate events, quality requirement analysis should focus on a broader area of defensible events to address the linkages among the disciplines. A formal structured analysis of both types of requirements can ensure sufficient coverage to both qualities.

Within requirements engineering, safety and security requirements are not always broadly considered beyond the single type of non-functional quality requirements. In addition, the following areas of requirements should be analyzed for effective safety and security:

- significant function, data, and interface requirements
- constraints on functional requirements
- architecture and design constraints
- subsystem functions and constraints
- software requirements

Firesmith proposes a common process to ensure basic collaboration [Firesmith 2006]. A defensibility analysis team with membership from both the safety and security teams should participate in system analysis, stakeholder analysis, asset analysis, vulnerability analysis, event analysis, agent analysis, danger analysis, risk analysis, significance analysis, and defense analysis to assemble defensibility work products. These will describe defensibility-related requirements to be included in requirements identification, requirements analysis, and requirements validation with the requirements team. Instead of separate safety and security certifications and accreditations, the defensibility team would develop a defensibility certification and accreditation to address the blend of safety and security requirements appropriate to the software-intensive system to be developed.

A collaborative process for defensibility would accomplish the following:

- ensure close collaboration among safety, security, and requirements teams
- better integrate safety and security processes into the SDLC, especially the requirements process
- develop all types of safety- and security-related requirements not just non-functional types
- ensure the developed requirements have appropriate requirement properties

2.5 FOCUS ON RESILIENCY: A PROCESS IMPROVEMENT APPROACH TO SECURITY

Lisa Young introduced the SEI CERT[®] Resiliency Engineering Framework to address the organizational security challenge of identifying and managing risks that have the most potential to disrupt core business drives and impede mission survivability with limited resources [Young 2007]. Security must be viewed as a business issue owned by the organization. Security is an investment and an enterprise process that must be effectively measured and managed.

Instead of managing threats and vulnerabilities by applying technology-only solutions with no articulation of a desired state, organizations must manage impacts and consequences toward a clearly articulated desired state balancing security with enterprise cost and risk. This will provide operational resiliency, allowing an organization to sustain missions in the face of risks.

Operational resiliency is an emergent property. Resiliency must be built into assets, processes, and services through effective resiliency engineering. Organizations must establish processes that develop, implement, and manage the operational resiliency of business services, related business process, and assets associated with the service, to benefit from improved operational resiliency.

The Resiliency Engineering Framework is a process framework for resiliency engineering. It defines basic capability areas and provides guidelines for security and business continuity process improvement. The framework emphasizes vital linkages between security, business continuity, and IT operations. It addresses operational risk management through process management and establishes a capability benchmark. The resiliency engineering body of knowledge for the framework is based on an affinity analysis of 750 best practices in security, business continuity, and IT operations through a collaboration of business continuity experts from numerous U.S. financial institutions and CERT security expertise. The framework links resiliency engineering and process improvement to establish a process improvement model for resiliency engineering. The framework architecture consists of 24 capability areas focused on resiliency of people, information, technology, and facilities, in the context of services and business objectives.

Use of the framework approach provides

- greater efficiency of resilience activities
- objective benchmarking of resilience capabilities
- improved operational risk management

Version 1.0 of the framework is to be published this year. Additional information about the framework and its availability can be found at <http://www.cert.org>. The resiliency framework is further described in three technical notes by Caralli [Caralli 2007, 2006, 2004]. These are available at the URLs below:

<http://www.cert.org/archive/pdf/sustainoperresil0604.pdf>

<http://www.cert.org/archive/pdf/managinges0412.pdf>

<http://www.sei.cmu.edu/publications/documents/07.reports/07tr009.html>

2.6 GETTING STARTED WITH MEASURING YOUR SECURITY

The presentation by Moss and Rice emphasized the growing need to quantify information security assurance throughout the life cycle as systems move from disparate stove-pipes to network-centric dynamic environments. Security assurance measures can help in

- determining if information security defects are being identified sufficiently early in the life-cycle where they are cheaper to fix
- identifying and removing potential vulnerabilities in software systems and developing more secure design practices
- identifying and investigating trends that require corrective actions such as improvements in training and procedure revisions
- determining if systems comply with required operational controls

To effectively incorporate metrics, organizations need to establish effective security assurance processes and requirements.

- National Institute of Science and Technology (NIST) has published a *Guide for Security Certification and Accreditation of Federal Information Systems (SP 800-37)* [Ross 2004].
- *National Information Assurance Certification and Accreditation Process (NIACAP)* is documented in NSTISSE No 1000 [NSA 2000].
- *Department of Defense (DoD) Enterprise Information Assurance Certification and Accreditation Process (DIACAP)* is documented in DoD 8510 [DoD 2006].
- *ISO/IEC 15026: Information Technology – System and Software Integrity Levels and Software Assurance* provides guidance on security assurance processes and requirements [ISO 1998].

To effectively support defining and collecting meaningful metrics, an organization needs to have initiated and be maturing effective software engineering measurement practices, which provide a foundation for successful implementation of security assurance measures. Best practices for measurement must be applied, such as normalization to ensure comparisons are appropriate, and triangulation to confirm appropriate analysis of relationships. Good measurement practices begin small, measure process behavior as well as results, and gain management support through regular reporting.

Start with expansion of the project cost, schedule, quality, and growth measures to include security activities. Select a manageable set of security measures and add more as the process is refined through learning. Train data collectors to apply their metrics knowledge to security or train security staff to become data collectors. Incorporate security measures into the existing measurement activities to provide effective project visibility as part of the normal project life cycle.

The use of measurement best practices in the area of security assurance will help avoid information limitations and problems. Be sure to measure processes, not people. Even the appearance of measuring people will lead to fabricated information. Be sure to provide feedback to the data providers as well as information to the data users; missing feedback will lead to late or missing data as other needs press on the time of the data providers.

The Goal-Question-Metric (GQM) Approach is another best practice that can enhance the identification, planning, and execution of effective security assurance. Instead of arbitrarily applying compliance metrics, determine intended goals and link these with organizational goals. Look for ways to link security measures with existing measures to avoid more data collection for its own sake. For example, see how identified defects can link to security vulnerabilities.

3 Strengthening Ties between Process and Security

3.1 SECURITY BIRDS OF A FEATHER (BOF) AT SEPG 2007

Larry McCarthy of Motorola coordinated an open meeting of all conference attendees interested in discussing ways to promote stronger links between security and CMMI. Motorola has proposed sharing its experiences in applying selected security standards in specific projects. However, this must be recognized as a single organization's approach and not sufficiently robust for broad application without extensive piloting in other settings.

Several of the BOF attendees expressed a desire to have security inserted as a mandatory part of an organization's process capabilities. Before this can be considered, a shared perspective of a vision of how process should tie to security is needed. From this agreed-upon perspective, gaps with the current model can be identified and definitions of needed additions subsequently developed.

Safety and dependability share many of the same connections with process as security. The publication of an SEI technical note +*SAFE, V1.2 A Safety Extension to CMMI-DEV, V1.2* (CMU/SEI-2007-TN-006, p 19) suggests an approach for documenting the work done by Motorola and other organizations as they pioneer effective security and process connections [ADD 2007]. This approach exposes their experiences for broader use and evaluation.

Participants have suggested expanding the security focus to include other aspects of assurance to better align with other initiatives already underway.

3.2 NDIA SYSTEMS ASSURANCE GUIDEBOOK

The National Defense Industrial Association (NDIA) Systems Engineering Division Systems Assurance Committee (<http://tinyurl.com/222hvg>) is sponsoring the development of a guidebook to provide practical guidance to augment systems engineering practice for the DoD contractor community, academe, and other commercial and government partners. The guidebook will include a synthesis of existing organizational knowledge, recommendations, policies, and standards. Consistency with international standards and current best practices is planned. Stakeholder review and piloting will continue through September 2007.

3.3 DHS SOFTWARE ASSURANCE PROGRAM

The Department of Homeland Security (DHS) hosted, with the Department of Defense, a Software Assurance Forum in March 2007 in Washington, DC.

DHS continues to actively supporting the efforts of the SEI in developing the "Build Security In" Web site which can be found at <https://buildsecurityin.us-cert.gov/daisy/bsi/home.html>. This site is a repository of software security best practices, tools, guidelines, principles, and other resources for use by participants in every phase of the software development life cycle.

3.4 ISSEA SYSTEMS SECURITY ENGINEERING CMM

The International Systems Security Engineering Association (ISSEA) is a 501(c)(6) non-profit trade association for Security Professionals involved in all aspects of the security engineering life cycle, including policy, risk management, requirements, engineering, design, analysis, certification and accreditation, testing, operations, and management. ISSEA is dedicated to the advancement of systems security engineering as a defined and measurable discipline through the further development of its theory and practice. Therefore, its objectives are to

- promote and enhance the Systems Security Engineering Capability Maturity Model (SSE-CMM / ISO 21827) and its use in system software developments where appropriate [ISO 2002].
- promote mature security capability among system and software developers, service providers, operations managers, and risk management professionals; and ensure that security be integral to development, integration, test, fielding, and operation of a system throughout its life cycle
- provide educational and networking opportunities to the systems security engineering community

ISSEA is running an ongoing effort to identify opportunities to collaborate with other initiatives for aligning with SSE CMM to promote mature security capability among system and software developers.

3.5 ISO/IEC 15026 “SYSTEMS AND SOFTWARE ASSURANCE”

In May 2007 ISO/IEC JTC1 SC7 approved the changes in 15026 “Systems and Software Assurance” to provide a process focus relevant to the needs of security [ISO 2007]. “System and software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycles.”

The draft standard, which has been released for comments, focuses on the assurance case that requires evidence sufficient to support arguments to justify claims. The general requirements revolve around a project’s establishing and maintaining an assurance case. The project shall ensure the following:

- Goals and objectives for safety, security, dependability and any other designated critical properties are formulated.
- Product assurance-related objectives, properties, or characteristics are explicitly selected for special attention and application of this standard to address the goals and objectives.
- Requirements for the achievement of these objectives, properties, or characteristics are defined.
- Measures for the requirements are selected and related to the desired characteristics.
- Criteria for the achievement or degree of achievement of these objectives, properties, or characteristics are selected and traced to requirements.

- Approaches for achieving the objectives, properties, or characteristics are planned, designed, and implemented, and mechanisms for demonstrating and documenting the results are defined.
- The extent of achievement is continuously monitored, documented, and communicated to stakeholders and managers.
- An assurance case documenting and communicating the extent of achievement is specified, developed, and maintained as an element of the system.
- The artifacts for documenting, analyzing, and communicating the required or claimed properties and characteristics and the extent of achievement are specified, developed, and maintained.
- Requirements of the approval authority are satisfied and necessary licenses or certifications are received.

Bibliography

URLs are valid as of the publication date of this document.

[ADD 2007]

Australian Department of Defense, Defence Materiel Organisation. *+SAFE, V1.2 A Safety Extension to CMMI-DEV, V1.2* (CMU/SEI-2007-TN-006). Software Engineering Institute, Carnegie Mellon University, 2007.
<http://www.sei.cmu.edu/publications/documents/07.reports/07tn006.html>

[Cappelli 2007]

Cappelli D., Trzeciak, R., & Moore, A. "Insider Threats in the SDLC." *Software Engineering Process Group Conference (SEPG 2007)*, Austin TX, March 2007. Software Engineering Institute, Carnegie Mellon University, 2007. <http://www.cert.org/archive/pdf/sepg500.pdf>

[Caralli 2007]

Caralli, R., Stevens, J., Wallen, C., White, D., Wilson, W., & Young, L. *Introducing the CERT Resiliency Engineering Framework: Improving the Security and Sustainability Processes* (CMU/SEI-2007-TR-009). Software Engineering Institute, Carnegie Mellon University, 2007.
<http://www.sei.cmu.edu/publications/documents/07.reports/07tr009.html>

[Caralli 2006]

Caralli, R. *Sustaining Operational Resiliency: A Process Improvement Approach to Security Management* (CMU/SEI-2006-TN-009, ADA446757) Software Engineering Institute, Carnegie Mellon University, 2006. <http://www.cert.org/archive/pdf/sustainoperresil0604.pdf>

[Caralli 2004]

Caralli, R. *Managing for Enterprise Security* (CMU/SEI-2004-TN-046, ADA430839). Software Engineering Institute, Carnegie Mellon University, 2004.
<http://www.cert.org/archive/pdf/managinges0412.pdf>

[DoD 2006]

Department of Defense. *Department of Defense Enterprise Information Assurance Certification and Accreditation Process (DIACAP)* documented in DoD 8510. Available through <http://iase.disa.mil/ditscap/> (2006).

[Firesmith 2007]

Firesmith, Donald. "Engineering Safety- and Security-Related Requirements for Software-Intensive Systems." *Software Engineering Process Group Conference (SEPG 2007)*, Austin, TX TN, March 2007. Software Engineering Institute, Carnegie Mellon University, 2007.
Available through <http://www.sei.cmu.edu/programs/acquisition-support/presentations.html>

[Ibrahim 2004]

Ibrahim L., Jarzombek, J., Ashford, M., Bate, R., Croll, P., Horn, M., LaBrayers, L., Wells, C., & Members of the Safety and Security Extensions Project Team. *Safety and Security Extensions for Integrated Capability Models*. U.S. Federal Aviation Administration, 2004.
http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf

[ISO 2007]

International Organization for Standardization. *ISO/IEC JTC 1 N 8629: SC 7 Proposed New Work Item on "Revision of ISO/IEC 15026 – Systems and Software Engineering. Systems and Software Assurance."* Secretariat, ISO/IEC JTC 1, American National Standards Institute, 2007.
<http://isotc.iso.org/livelink/livelink/fetch/2000/2122/327993/755080/1054034/2541793/JTC001-N-8629.pdf?nodeid=6558690&vernum=0>

[ISO 2002]

International Organization for Standardization ISO/IEC JTC 1/SC 27 N3416: Notice of Publication of the International Standard ISO/IEC 21827: Information Technology –Systems Security Engineering – Capability Maturity Mode (SSE-CMM) ISSEA International Systems Security Engineering Association, 2002. <http://www.issea.org/docs/27n3416.pdf>

[ISO 1998]

International Organization for Standardization. *ISO/IEC 15026: Information Technology – System and Software Integrity Levels and Software Assurance*, 1998. Available through
<http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=26236>

[Moss 2006]

Moss, Michele. "Getting Started with Measuring Your Security." *Tenth Annual PSM Users' Group Conference*. Vail CO, July 2006. Practical Software and Systems Measurement,
http://www.psmc.com/UG2006/Presentations/13_GettingStartedwithSecurityMeasurement_Moss.pdf (2006).

[NSA 2000]

National Security Agency. *National information Assurance Certification and Accreditation Process (NIACAP)*, 2000. Available through <http://www.stormingmedia.us/60/6041/A604193.html>

[Ross 2004]

Ross, R., Swanson, M., Stoneburner, G., Katzke, S., & Johnson, A. *Guide for Security Certification and Accreditation of Federal Information Systems*. National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-37/SP800-37-final.pdf>

[SEI CERT 2007]

Software Engineering Institute CERT *Cylab Common Sense Guide*. Carnegie Mellon University, 2007. http://www.cert.org/insider_threat

[van Wyk 2007]

van Wyk, Kenneth. "Software Security – Setting the Stage." *Software Engineering Process Group Conference (SEPG 2007)*, Austin TX, March 2007. Software Engineering Institute, Carnegie Mellon University, 2007. <http://www.sei.cmu.edu/sepg/2007/pdf/top10/vanwyk.pdf>

[Young 2007]

Young, L. "Focus on Resiliency: a Process Improvement Approach to Security." *Software Engineering Process Group Conference (SEPG 2007)*, Austin TX, March 2007. Software Engineering Institute, Carnegie Mellon University, 2007. <http://www.cert.org/archive/pdf/sepg0703.pdf>

The entries in this list were referenced by various speakers in the security track at SEPG 2007.
URL is valid as of the publication date of this document.

[Anderson 2001]

Anderson, Ross. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2001. <http://www.cl.cam.ac.uk/~rja14/book.html>

[Graff 2003]

Graff, Mark & vanWyk, Kenneth. *Secure Coding Principles and Practices*. O'Reilly, 2003.

[Hoglund 2004]

Hoglund, Greg & McGraw, Gary. *Exploiting Software: How to Break Code*. Addison-Wesley, 2004.

[Howard 2002]

Howard, Michael & LeBlanc, David. *Writing Secure Code, Practical Strategies and Proven Techniques for Building Secure Applications in a Networked World*. Microsoft Press, 2002.

[Howard 2006]

Howard, Michael & Lipner, Steve. *The Secure Development LIFECYCLE: A Process for Developing Demonstrably More Secure Software*. Microsoft Press, 2006.

[Leveson 1995]

Leveson, Nancy. *Safeware System Safety and Computer: A Guide to Preventing Accidents and Losses Caused by Technology*. Addison-Wesley Publishing Company, 1995.

[Viega 2002]

Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley, 2002.

[Wiegers 1999]

Wiegers, Karl. *Software Requirements: Practical Techniques for Gathering and Managing Requirements Throughout the Product Development Cycle*. Microsoft Press, 1999.

| | | | | |
|---|--|---|---|---|
| REPORT DOCUMENTATION PAGE | | | <i>Form Approved</i> OMB No. 0704-0188 | |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave Blank) | | 2. REPORT DATE September 2007 | | 3. REPORT TYPE AND DATES COVERED Final |
| 4. TITLE AND SUBTITLE Process Improvement Should Link to Security: SEPG 2007 Security Track Recap | | | 5. FUNDING NUMBERS FA8721-05-C-0003 | |
| 6. AUTHOR(S) Carol Woody | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2007-TN-025 | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES | | | | |
| 12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS | | | 12B DISTRIBUTION CODE | |
| 13. ABSTRACT (MAXIMUM 200 WORDS) Security is a very visible issue these days for software. New software products are continuously reported to be vulnerable to attack and compromise; organizations must support an expensive unending update-and-upgrade cycle. Process improvement has been proposed as a mechanism for addressing security challenges, but the Capability Maturity Model Integration (CMMI®) approach does not specifically address security, so the linkages for the Software Engineering Process Group (SEPG) community are unclear. The security track at the SEPG 2007 conference was developed to provide a forum for identifying the appropriate ties between process improvement and security. This document summarizes the content shared at the conference and identifies several subsequent steps underway toward strengthening those ties. | | | | |
| 14. SUBJECT TERMS software security, systems assurance, software assurance, insider threat, resiliency, SEPG 2007, security track | | | 15. NUMBER OF PAGES 37 | |
| 16. PRICE CODE | | | | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL | |